Model-Based Application: The Galen Structured Clinical User Interface

Laurence Alpay[1], Anthony Nowlan[3], Danny Solomon[4], Christian Lovis[2], Robert Baud, Tony Rush[3], Jean-Raoul Scherrer[1]

[1] Medical Informatics Centre, Geneva Hospital, Geneva, Switzerland

[2]Medicine Department, Geneva Hospital, Geneva, Switzerland

[3] Medical Products Group, Hewlett Packard Ltd, Bristol, U.K

[4] Medical Informatics Group, University of Manchester, Manchester, U.K

Researchers in the Artificial Intelligence in Medicine community are facing the challenge to design and develop clinical systems which are intuitive to use and adequately expressive. This paper reports on a Structured Clinical User Interface (SCUI) which achieves this goal by using technologies developed in the Galen project: a new formalism to represent models of terminology coupled with a server to access and use this knowledge. This approach opens the way to the development of clinical systems which use conceptual knowledge, driven by models of terminology, in a dynamic and flexible way. Furthermore, the overall task of building a clinical application is separated into a terminological part handled by the server and an application part handled by application developers without needing to worry about implementing the terminology. The SCUI is specifically developed and tested in the context of infectious diseases to satisfy the demands made by the medical intensive care unit to the Geneva Hospital's microbiology laboratory.

1 Introduction

The SCUI is developed in the context of the European AIM project Galen. The purpose of Galen is to establish the foundations for the next generation of coding systems to support integrated clinical information systems. The heart of Galen is a Semantical Encyclopaedia of Terminology (SET). The SET consists of a generative Master Notation (also referred to as GRAIL for Galen Representation and Integration Language) and a Coding Reference (CoRe) model for medical terminology formulated in that notation [1], [2] and information modules providing multilingual lexicons, conversions to and from existing coding systems. The SET is encapsulated by an application programming interface to form a general Terminology Server (TeS) for use by other applications in a client-server environment [3], [4]. The SCUI is a test bed for the Galen technologies; and as such it is aimed to provide input (to the requirements) for the bottom up development of the Master Notation and the CoRe model, and to evaluate the effectiveness of this notation within a clinical setting. Furthermore, the SCUI [5] is initially implemented as a bilingual system (French and English) to provide a means of testing the multilingual module of the TeS [6].

2 Context of the SCUI

2.1 Clinical Setting

The SCUI is intended to collect clinical information of the infectious disease domain in a large university hospital like the Geneva Hospital. We have specified *nosocomial infections* within the infection diseases specialty as the domain of application. Nosocomial infections or hospital infections are best defined as infections which are neither present nor incubating at the time of the admission to the hospital, and which are usually developed at least 48 hours following the patient's admission [7], [8]. Detailed definitions for various types of nosocomial infections have been established by the Centers for Disease Control (CDC) in Atlanta in 1988 [9]. Such infections are relevant since they increase patient morbidity and mortality as well as hospital costs [8], [10]. Being acquired in the hospital, they represent a major objective for the development of surveillance systems [11]. Within the domain of infectious diseases, we have focused on the subdomain of urinary infections. There are a number of reasons for choosing this infectious subdomain. First, there is a strong interest from clinicians for this type of infection. In addition, urinary infectious diseases are very frequent and easy to detect. Furthermore, urinary infections are typical nosocomial infections and represent an additional problem to the patient's morbidity. Finally, the symptomatology of urinary infections, as well as the necessary laboratory tests, are well known.

We have chosen the *medical intensive care unit* (MICU) as the target ward for the development of the demonstrator. Patients hospitalised in intensive care units are at a much higher risk of developing nosocomial infections than in more general wards: infections acquired in these wards account for more than 20% of nosocomial infections, although these care for less than 10% of hospitalised patients [12]. The urinary tract as an infection site counts for 24% and reasons for higher infection rates in intensive care units include, among other things, urinary bladder catheterization [13].

2.2 Design Considerations

The development of the SCUI is centered around requests made to the microbiology laboratory by the MICU. The initial aim of the SCUI is to improve requests made by the MICU by asking significant and relevant clinical information about the patients [14]. The current procedure at the Geneva Hospital for ordering a microbiology examination is as follows: 1) the health care provider in the MICU makes a microbiology request; 2) the request goes through the Geneva Information System for laboratory exams called UNILAB [24] (a part of the DIOGENE Hospital Information System [15]); 3) the request arrives at the laboratory, is dispatched, and investigations are made until results are obtained; 4) results are collected, validated, and returned to the clinical service; 5) the clinical service interprets the results, takes action following the test results (e.g. treatment, no treatment, other investigations) and may make a new request.

This current procedure has some drawbacks in that some clinical information (like state of immunity, state of illness, presence of fever) which would be useful to the laboratory is usually missing. Morever, the data collected during a microbiology laboratory request are not analyzed further, so that the clinical staff can ask for additional information. With the SCUI, the requests made to the microbiology laboratory should be improved: On one hand, the microbiology laboratory needs to gather relevant clinical information with the test requested in order to increase the chance of identifying an infectious agent. On the other hand, the clinician aims at receiving a better and faster service from the laboratory in the process of microbiological test requests.

2.3 Clinical Information Requirements

The SCUI is applicable whenever a urinary infection is suspected in the clinical ward. The types of data to be collected while making a request to the laboratory have been grouped into the following sections. It is much more detailed than the information usually presented and is an important aim to make detailed patients' information available to the user where necessary:

1) GENERAL INFORMATION e.g. the patient's mobility, consciouness, dependency;

2) URINARY HISTORY e.g. the patient's past urinary infections and any related treatment, as well as information about any event of lithiasis and nephrectomy;

3) SIGNS AND SYMPTOMS e.g. general findings such as fever, chills; specific findings such as alguria, dysuria, haematuria;

4) DIAGNOSTIC INFORMATION e.g. information on the diseases related to urinary infections such as cystitis, cystopyelitis, pyelonephritis, renal abscess, etc;

5) PREVIOUS LABORATORY RESULTS e.g. under the form of interpreted information about hematology, information about the urine such as the estimation of the urine red cells, the urine white cells, the urine glucose, the urine pH and the urine proteins;

6) SAMPLE COLLECTION e.g. general information on the sample taken such as the date when it was taken, the type of examination, the type of sample, the operator, the place where the sample was taken (the emergency room, the operating room); information on the justification of the exam such as monitoring; information on the sample mode used such as direct mode, permanent urine bladder catheter, transitory urine bladder catheter, or renal catheter; information on the status of the catheter, i.e. clean or dirty, whether it is old or new, and so on.

The SCUI application guides the user into collecting specific detailed information, that is, to use the terminology in the CoRe model which is required in order make a microbiological laboratory request. In other words, the CoRe model contains the medical knowledge needed to collect this detailed patient's data (see Section 3.2). The SCUI makes use of the Terminology Server to access the CoRe model. The server provides the application with a set of terminological services such as conceptual and multilingual (see Section 3.3). The detailed clinical information collected from the user is then stored into the SCUI patients' medical records. Once the request has been made, it is available to the microbiology laboratory.

3 Terminology Server

3.1 Terminology Modelling and Representation

The building of a Terminology Server, which is able to conciliate a number of classifications and nomenclatures, requires a powerful knowledge representation schema and accordingly a representation language. GRAIL is such a language, primarily intended to model terminology. It is used in the development of medical domains, and one of the final goals is to design a large medical model in the form of a semantic network in order to accommodate all sensible medical expressions. Two processes are available for this task. First, a sanctioning mechanism allows the specification of what is medically sensible to say, i.e. what is not sanctioned is not accepted. Second, an indefeasible definitional schema is ready to grasp the reality in the model. Any expression in the network is later made available to the outside world in a canonical form [1]. For example, the concepts nephritis and oliguria are defined as follows in the CoRe model, where *which* and *name* are two GRAIL key words: *which* to introduce the definition of a new concept, as a specialization of existing concepts and *name* allows to name that concept.

(InflammatoryProcess which hasLocation Kidney) name Nephritis.

(Micturition which hasFeature (ProcessActivity which hasState Decreased)) name Oliguria.

3.2 Conceptual Models

Given the clinical requirements described in Section 2.3, the SCUI needs to use a variety of concepts. This means, for example, that the TeS needs to know about urinary devices, signs and symptoms (specific and general) as well as geographical locations (e.g. operating room) and actors (e.g. nurse, doctor). In other words, the TeS is to tell the application what can be sensibly and necessarily said about a concept like dysuria, to support the SCUI interface to help the clinician to enter the information. The GRAIL formalism makes a separation between the definition of a concept and its properties. For example, for the concept dysuria, the TeS will obtain from the CoRe model the definition of dysuria, i.e. pain which occurs during micturition (micturition being itself a complex concept), and properties of dysuria like its causal agent and the types of bacterium that causes it. The collection of clinical information for dysuria by the user is then structured and driven by what comes from the model. Furthermore, the data collection can be made more or less detailed according to what the end-user wants to express about some particular clinical information.

Although it is expected that the CoRe model will be able to support the terminological needs of the application, it is acknowledged that the application has terminology requirements which remain “local” to the application. These local requirements are twofold. First, the terminological extension model i.e. extra detailed concepts specific to the application and which are not found in the Core model. Second, the dialogue model which uses the terminology as a “hook” to add extra information like what concepts are interesting in a particular section of the SCUI application. This results in a situation where there is a terminology “linkage” between the CoRe model and the SCUI specific model. For instance, the CoRe model will be able to provide the application with sensible statements of what can be said about dysuria, while the SCUI model will specify that dysuria should be asked for in the “Signs and Symptoms” section of the application.

3.3 Services of the Terminology Server

The Terminology Server [3], [4] can be viewed as a networked interface to a set of terminological services. The TeS contains different modules, each of which provides a specific set of services to clinical applications, for example the SCUI (see Figure 1.).The *Concept Module* (CM) [16] provides GRAIL concept services: it allows the navigation and manipulation of expresssions in the CoRe model to answer questions like, for example, “what can be said about dysuria”.

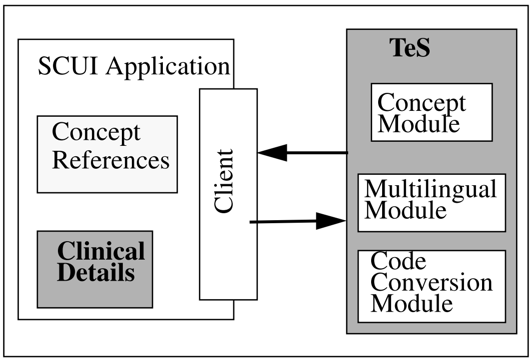

Figure 1. Connection SCUI & TeS

Figure 2. SCUI Menus

The *Multilingual Module* (MM) [6] provides naming services: it generates natural language (e.g. mostly in English and in French in the current version) from expressions in the CoRe model. For example, from the expression: “Ulcer which hasLocation Stomach” the generator will produce “stomach ulcer” in English and “ulcère de l’estomac” in French. This module provides the application with a language service, particularly useful when developing a single application to be used in more than one natural language context (see Section 4.3). The *Code Conversion Module* (CCM) [17] provides coding services: it relates concepts of the CoRe model to codes in existing systems such as ICD-9 or Snomed. For instance, the CCM can answer the request “return the code for the concept: Ulcer which hasLocation Stomach” with ICD-9 531.1.

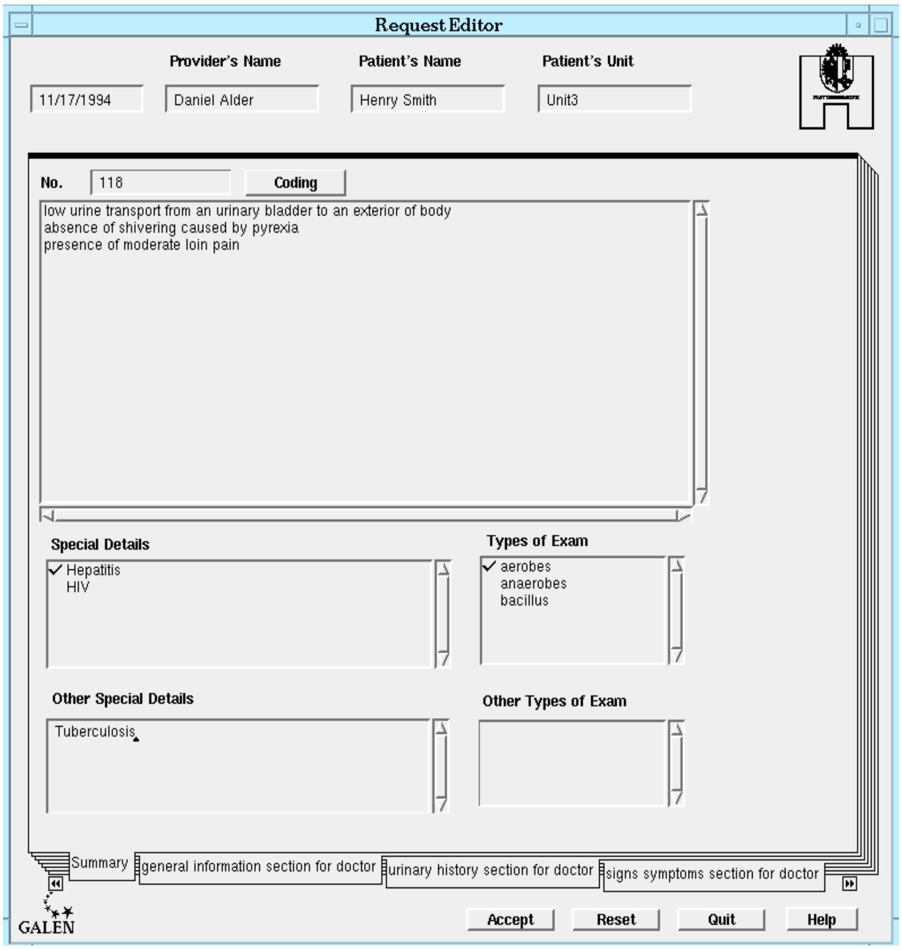

Figure 3. Example of the main page of the request editor while a request is being made

The main page of the request editor includes: a patient location pane at the top; the request number with the set of data which has been collected so far by the user and generated in natural language; the option to view codes for these data; special details and types of exam to be selected from given lists or to enter as free text into a text editor. The buttons to access the clinical sections and the current summary pane are shown on a slider. By clicking on a section name, the model-based interface for that section appears (see Figure 4). Control buttons to save or dismiss the request are at the bottom of the window.

4 SCUI Application

4.1 Overview

In order to run the SCUI, the application needs to establish a connection with the TeS. A *connection session* between the application and the TeS consists of: connecting to the server, performing a series of requests (e.g. specify a terminological operation like “specialise relationships” to get the attributes that are used to specialise a given concept), and disconnecting from the TeS. By connecting to the server, the application becomes its client (see Figure 1.). The TeS which runs as a network server, communicates with the SCUI client using TCP/IP over sockets [16]. The SCUI client is provided with a library which is used to interact with the server. The SCUI is developed in Smalltalk/VisualWorks^{TM} .

We have separated the overall task of building the clinical application into two: a) a terminological part that the TeS handles and b) an application part which can be built by the application builders without needing to worry about implementing the terminology. This separation constitutes a new approach in the task of developing clinical applications. The clinician or the nurse starts the SCUI application by giving his/her user name and password. From then on, the clinician interacts mostly with the application using the mouse. A set of menus (see Figure 2) is available to the user. For example, the user can enter a new SCUI request (see Figure 3), or visualise a previous request and possibly restart a request which was not complete. The user can also access administrative information such as the patient’s personal details and the patient’s SCUI medical record, and make use of configurational options (e.g changing font or language) found under the Special menu (see [18] for details).

The SCUI connects to the TeS whenever, depending on the user’s action, it needs to access concepts from the CoRe model and in a given language. Figure 3 shows the main page of the request editor where a set of clinical information has already been collected. These data are presented in a natural language form generated from the MM of the server. To get the model of a specific section (see Figure 4), the user clicks on the button with the section name on it e.g. signs symptoms section.

4.2 Model-Based Interface

There are two important aspects to stress here about the interface related to the conceptual models (found in the request editor window). First, the interface construction for each section of the SCUI scenario (see Section 2.3) is not fixed but rather is constructed *dynamically* based upon what comes from the CoRe model via the TeS. This is an advantage of the system over traditional form-based interfaces. The application is much more flexible than traditional systems and much better tailored to the particular clinical situation as it has access to a server of terminology rather than to static terminology. This means that if a new concept which is of interest to the SCUI is entered in the CoRe model, it will appear in the corresponding section.

Second, when making a request, the clinician clicks on the various concepts and associated attributes which are available to him/her and which are relevant for the given patient. By doing so, the clinician constructs simple or complex concept entities to express and detail the patient's clinical information. For example, he/she can express the fact that a loin pain has a moderate severity (LoinPain whichHasSeverity Moderate in GRAIL), and thus builds GRAIL expressions. Figure 4 shows an example of the model-based interface of the request editor for the "Signs Symptoms Section" of the SCUI application. The user clicks on the concepts and their corresponding values e.g. rigor and absence.

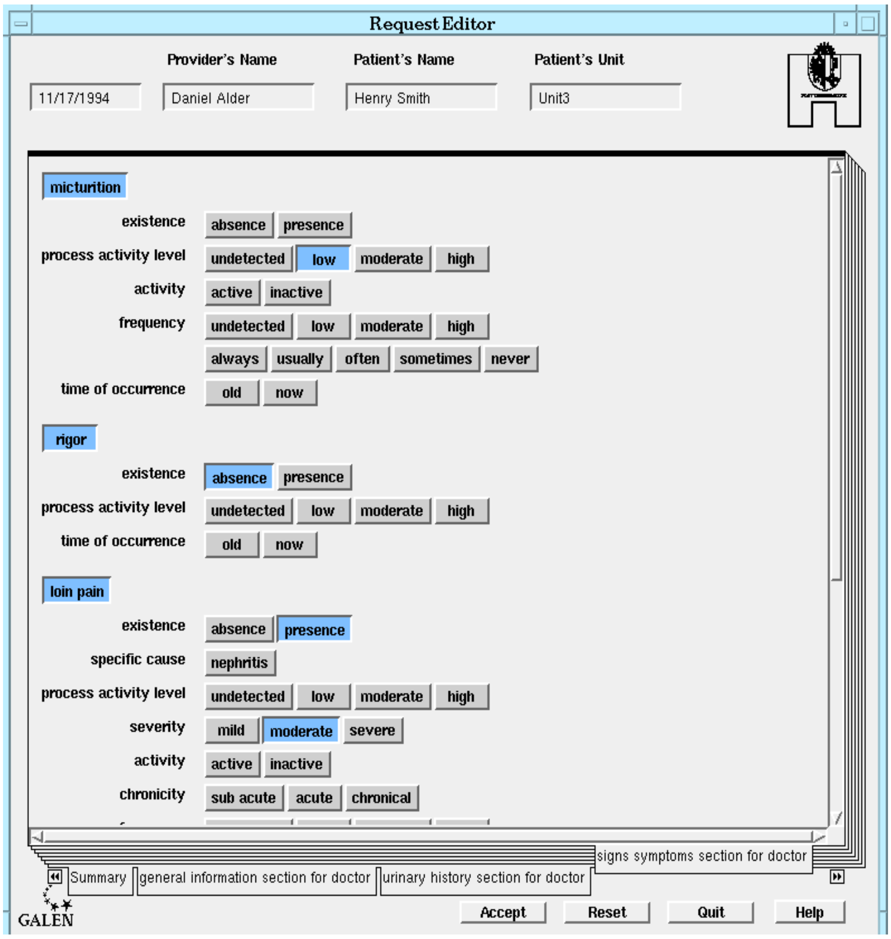

Figure 4. Example of the model-based interface of the request editor

Figure 4 shows elements of the model for the "Signs Symptoms Section" of the SCUI application. Patient's detailed clinical data are collected by selecting elements of the GRAIL model with the mouse. Each concept like rigor is associated with one or more relations like existence and its corresponding values e.g. presence, absence. The data collected and generated in natural language are shown in the summary pane (as illustrated in Figure 3).

4.3 Other Particularities

Multilingual Feature.

The work carried out on the MM module of the TeS has focused on the generation for English and French. Preliminary experiments with other languages (such as German, Dutch, Finnish, Italian and Spanish) have started for the generation of noun phrases**.** The results are encouraging. Figure 5 shows the screen dumps for the phrase "presence of acute cystitis" as generated by the MM in various languages and which is displayed on the summary pane of the request editor window.

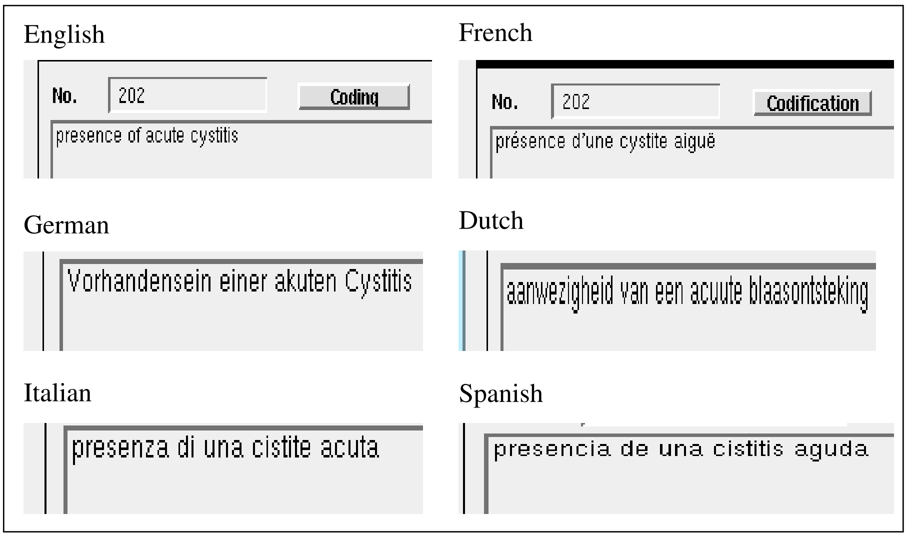

Figure 5. Examples of the use of TeS multilingual services for the SCUI application

The noun phrase "presence of acute cystitis" generated by the MM of the TeS in various European languages and as it appears to the end-user on the summary pane of the request editor.

Types of End-Users.

As part of testing the Galen technologies, the SCUI application has the option to view different models according to the kinds of clinical users i.e. a nurse or a doctor who is using the application. This switch is important to illustrate the flexibility of the application to use different conceptual models according to the kind of users who is running the application. In the current prototype, the nurse has only four clinical sections against the doctor's six, and the contents of all the clinical sections are different: there are concepts which appear on both forms and also on one but not the other. The model shown in Figure 4 is for a user-doctor, and the section names in Figures 3 and 4 also specified that there are for a user-doctor.

Tools for customizing the clinical sections.

The SCUI includes a tailoring tool which allows the clinician to customize the contents of the clinical sections according to his/her needs [18]. In particular, the aim of this tool is let the user add and remove concepts from a clinical section, and to create a new section if necessary. This tool makes use of a complementary tool called Confuser (CONcept Finder for USERs) which contains a series of functionalities to browse through the CoRe model [23].

5 Prototypes and Evaluation

The development of the SCUI has been going through a cycle of "prototyping and testing". The expectations of the first version described in [19] were to explore the modelling process and make use of some of the Terminology Server services such as the multilingual module. The work for the first version produced requirements for an "intelligent" dialog. The current version (version 2) has evolved in parallel with the development of the Tes modules and of the models. The application is run multilingual: annotations of concepts used by the SCUI have been done in English and French, and that task has started for other languages like German (see Section 4.3). Feedback and comments on the prototypes have been obtained from doctors at the Geneva Hospital and within the Galen consortium. Larger scale user evaluation is expected for the current version SCUI in the near future.

6 Conclusions

The development of the SCUI stems from a previous experience [20], [21]. In a similar line of research, the SCUI presents an innovative approach in the design and development of clinical systems: it not only has features which are not usually found in other traditional systems (in regards to its dynamic interface construction), but also has access to new technologies (through a range of terminological services).

Key aspects relevant to the SCUI include the following: First, the application uses a separate generic server which provides terminological services (e.g. multilingual) and which can be used by other applications simultaneously. This corresponds to a new approach in the development of clinical systems and the sharing of medical knowledge [22]. Second, the conceptual models which the SCUI makes use of are developed with the new formalism GRAIL for representing conceptual knowledge. Third, the SCUI exploits new technology for model-based interface applications, and finally, the application deals with collecting patients' detailed clinical information. Although the SCUI is still at the prototype stage, it provides promising grounds and is an illustration of how the Galen technologies can be used. Moreover, the SCUI offers a framework for the construction of practical and usable medical applications in the future.

Acknowledgements

The current developments, carried out at the Geneva University Hospital, are part of the GALEN project (A2012) and have been made possible thanks to the funding of the AIM programme of the EU, for which the Swiss government (CERS) has given full support. Galen documents are available from the Medical Informatics Group of Manchester.

7 References

[1] AL Rector, WA Nowlan, A Glowinski, G Matthews. *The Master Notation.* Galen Deliverable 6, March 1993.

[2] AL Rector, WA Nowlan, S Kay. *Conceptual Knowledge: The CoRe of Medical Information Systems,* MEDINFO 92, K C Lun et al. editors, North Holland, pp1420-1426.

[3] TW Rush, W Claassen, I Piumarta, P Pomper,WA Nowlan, P Zanstra. *Consolidated Specification of the Terminology Server and PUPPI.* Galen Deliverable 9, June 1993.

[4] AL Rector , WD Solomon, WA Nowlan, TW Rush. *A Terminology Server for Medical Language and Medical Information Systems.* IMIA Working Group 6 on Natural Language and Medical Concept Representation. Vevey, Switzerland, May 94.

[5] LL Alpay, C Lovis, WA Nowlan, WD Solomon, TW Rush, RH Baud, C Juge and J-R Scherrer. *Constructing Clinical Applications: The Galen Approach.* Accepted to MEDINFO95, Vancouver, Canada.

[6] J Wagner, A-M Rassinoux, R Baud, J-R Scherrer. *Generating Nouns Phrases from Medical Knowledge Representation.* MIE 94, Lisbon, Portugal, May 94.

[7] RL Thompson. *Surveillance and reporting of nosocomial infections*. In: Prevention and Control of Nosocomial Infections. Wenzel RP (ed.), Williams & Wilkins, Baltimore, 1987, pp. 70-82.

[8] G Ayliffe, E Lowbury, A Geddes, J Williams. *Control of Hospital Infection.* A Practical Handbook. Chapman & Hall Medical, London, 1992, pp. 1-11.

[9] J Garner. *CDC definitions for nosocomial infections, 1988.* J. Infect. Control 16 : 128-140, 1988.

[10] W Schaffner. *The global impact of hospital-acquired infections.* In: Prevention and Control of Nosocomial Infections. Wenzel RP (ed.). Williams & Wilkins, Baltimore, 1987, pp. 13-18.

[11] R Wenzel, S Streed. *Surveillance and use of computers in hospital infection control.* J. Hosp. Infect. 13 : 217-229, 1989.

[12] D Pittet, L Herwaldt, R Massanari. *The intensive care unit.* In: Hospital Infections. Bennett JV, Brachman PS (eds.). Little, Brown, and Company, Boston, 1992, pp. 405-439.

[13] R Weinstein. *Epidemiology and control of nosocomial infections in adult intensive care units.* Am. J. Med. 91, suppl 3B, 179-184, 1991.

[14] E Safran, L Alpay, C Lovis, R Baud, J-R Scherrer. *User Centred Design and Requirements for and Specification of Structured Clinical User Interface (SCUI).* Galen Deliverable 5, November 1992.

[15] Scherrer J-R, Baud RH, Hochstrasser D, Ratib O. *An Integrated Hospital Information System.* MD Computing, 7, 81-89, 1990.

[16] WD Solomon, TW Rush. *The consolidated Terminology Engine and SET Tools and Terminology Server version 2.* GALEN Deliverable 15, March 1994.

[17] F Lorino, A Rossi Mori, P Agnello, E Galeazzi. *The Consolidated CoRe model and Code Conversion Information Module.* GALEN Deliverable 16, January 1994.

[18] L Alpay. *SCUI User Guide.* Galen Documentation. October 1994.

[19] L Alpay, C Lovis, J Wagner, P-A Michel, R Baud. *SCUI version 1 and Evaluation.* Galen Deliverable 11, November 1993.

[20] WA Nowlan, AL Rector, C Goble, B Horan, T Howkins, A Wilson. *PEN & PAD: a Doctor's Workstation with Intelligent Data Entry and Summaries.* SCAMC 90, RA Miller (ed), IEEE Press, Washington, pp941-942.

[21] WA Nowlan, AL Rector, S Kay, B Horan, A Wilson. *A Patient Care Workstation Based on User Centred Design and A Formal Theory of Medical Terminology: PEN & PAD and the SMK Formalism.* AMIA 1992, pp855-857.

[22] WA Nowlan, AL Rector, TW Rush, WD Solomon. *From Terminology to Terminology Services.* SCAMC 94, J.G. Ozbolt (ed), Hanley & Belfus publishers, Philadelphia, pp150-154.

[23] S Bechhofer. *Confuser User Guide.* Galen Documentation. July 1994.

[24]Lagana M, Berney J-P, Schulthess P, Nawrocki B. *Guide Utilisateur UNILAB.* Internal Report. September 1993.